



TRANSMITTAL FORM

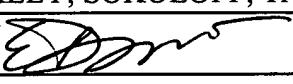
(to be used for all correspondence after initial filing)

		Application No.	09/927,250
		Filing Date	August 10, 2001
		First Named Inventor	Mike J. Little
		Art Unit	2173
		Examiner Name	Nguyen, Cao H.
Total Number of Pages in This Submission	48	Attorney Docket Number	5166P006

ENCLOSURES (check all that apply)

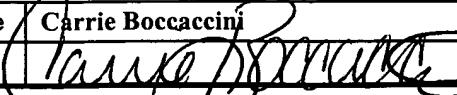
<input checked="" type="checkbox"/> Fee Transmittal Form <input checked="" type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment / Response <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> PTO/SB/08 <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/ Incomplete Application <input type="checkbox"/> Basic Filing Fee <input type="checkbox"/> Declaration/POA <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s)	<input type="checkbox"/> After Allowance Communication to Group <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input checked="" type="checkbox"/> Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input type="checkbox"/> Other Enclosure(s) (please identify below): <div style="border: 1px solid black; height: 40px; width: 100%;"></div>	
			<input type="checkbox"/> After Allowance Communication to Group
			<input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences
			<input checked="" type="checkbox"/> Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
			<input type="checkbox"/> Proprietary Information
			<input type="checkbox"/> Status Letter
			<input type="checkbox"/> Other Enclosure(s) (please identify below): <div style="border: 1px solid black; height: 40px; width: 100%;"></div>
			<input type="checkbox"/> Postcard
			<input type="checkbox"/> Remarks
			<input type="checkbox"/> Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Elena B. Dreszer, Reg. No. 55,128 BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Signature	
Date	7-21-04

CERTIFICATE OF MAILING/TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Typed or printed name	Carrie Boccaccini
Signature	
Date	July 21, 2004



FEES TRANSMITTAL for FY 2004

Effective 01/01/2004. Patent fees are subject to annual revision.

Applicant claims small entity status. See 37 CFR 1.27.

TOTAL AMOUNT OF PAYMENT (\$ 330.00)

Complete if Known	
Application Number	09/927,250
Filing Date	August 10, 2001
First Named Inventor	Mike J. Little
Examiner Name	Nguyen, Cao H.
Art Unit	2173
Attorney Docket No.	5166P006

METHOD OF PAYMENT *(check all that apply)*

<input checked="" type="checkbox"/> Check	<input type="checkbox"/> Credit card	<input type="checkbox"/> Money Order	<input type="checkbox"/> Other	<input type="checkbox"/> Non
<input type="checkbox"/> Deposit Account				

Deposit Account Number	02-2666
Deposit Account Name	Blakely, Sokoloff, Taylor & Zafman LLP

The Commissioner is authorized to: (check all that apply)

Charge fee(s) indicated below Credit any overpayments

Charge any additional fee(s) or underpayment of fees as required under 37 CFR §§ 1.16, 1.17, 1.18 and 1.20.

Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account

FEE CALCULATION

1. BASIC FILING FEE

Large Entity	Small Entity				
Fee Code	Fee (\$)	Fee Code	Fee (\$)	<u>Fee Description</u>	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

2. EXTRA CLAIM FEES

Total Claims		Claims below	Fee Paid
Independent Claims	- 20** =	X	
Multiple Dependent	- 3 =	X	

Large Entity		Small Entity		
Fee Code	Fee (\$)	Fee Code	Fee (\$)	<u>Fee Description</u>
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple Dependent claim, if not paid
1204	86	2204	43	**Reissue independent claims over original patent
1205	18	2205	9	**Reissue claims in excess of 20 and over original patent

***or number previously paid, if greater. For Reissues, see below.*

SUBMITTED BY

Name (Print/Type)	Elena B. Dreszer	Registration No. (Attorney//Agent)	55,128	Telephone	(408) 947-8200
Signature				Date	7-21-09

Based on PTO/SB/17 (10-03) as modified by Blakely, Solokoff, Taylor & Zafman (wlr) 02/10/2004.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450



AF/2173
7/21/04

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 09/927,250

Confirmation No.: 4304

Inventors: Mike J. Little, et al.

Filed: August 10, 2001

Art Unit: 2173

Examiner: Nguyen, Cao H.

Docket No.: 5166P006

Customer No.: 08791

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

on July 21, 2001
Date of Deposit

Carrie Boccaccini
Name of Person Mailing Correspondence

Signature

Carrie Boccaccini
Date

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450.

APPEAL BRIEF
IN SUPPORT OF APPELLANTS' APPEAL
TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Sir:

This Brief is submitted in triplicate in support of this appeal from a final decision of the Examiner, mailed February 24, 2004. Consideration of this appeal by the Board of Patent Appeals and Interferences for allowance of the above-captioned patent application is respectfully requested.

07/28/2004 SMINASSI 00000010 09927250

01 FC:1402

330.00 OP

TABLE OF CONTENTS

I.	REAL PARTY IN INTEREST	1
II.	RELATED APPEALS AND INTERFERENCES.....	1
III.	STATUS OF CLAIMS.....	1
IV.	STATUS OF AMENDMENTS	1
V.	SUMMARY	2
	A. Summary of Invention	2
	B. Summary of Rejections	2
	C. Summary of the References.....	3
VI.	ISSUES	3
VII.	GROUPING OF CLAIMS	3
VIII.	ARGUMENT	4
A.	The 35 U.S.C. 102(b) rejection of claims 1-29 was erroneous, because Bach fails to teach each and every limitation of independent claims 1, 12, and 23	4
1.	Bach is silent regarding a CLI description file for each command in the set of commands, and therefore fails to teach the feature of “creating a CLI description file for each command in the set of commands to abstract the CLI from the system” as required by independent claims 1, 12, and 23.....	4
2.	The language in Bach relied upon by the Office Action relates to objects framework and to graphical user interface (GUI), and thus is unrelated to the feature of “creating a CLI description file for each command in	

the set of commands to abstract the CLI from the system" of claims 1, 12, and 23 13.....	5
IX. CONCLUSION.....	6
IX. APPENDIX A (37 C.F.R. § 1.192 (c)(9))	A1

I. REAL PARTY IN INTEREST

The real party in interest is Occam Networks, Inc., a Delaware corporation having a place of business at 77 Robin Hill Road, Santa Barbara, Ca., 93117.

II. RELATED APPEALS AND INTERFERENCES

Appellants are not aware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-29 are currently pending, have been finally rejected and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

Claims were amended in response to the Office Action mailed on September 15, 2003. The amendments have not been acted upon by the Examiner. The amendments were introduced by the Applicants to merely highlight, without altering, the meaning of the claim language and therefore do not necessitate a new search. For example, the feature of “creating a CLI description file for each command in the set of commands, the CLI description file mapping the command with the action and the associated objects and the methods” has been replaced with “creating a CLI description file for each command in the set of commands to abstract the CLI from the system, the CLI description file mapping the command to be used with the CLI with the action and the associated objects and the methods”. Because the addition of the “to abstract the CLI from the system” language merely highlights the purpose of creating a CLI description file for each command in the set of commands, without altering the meaning of such operation, no additional search is required and thus this amendment, as well as the rest of the amendments should be entered.

V. SUMMARY

A. Summary of Invention

The present invention addresses the need for providing a generalized user interface, such as a command line interface (CLI), to an embedded system in such a way that the generalized user interface is decoupled from the services and features delivered and implemented in the embedded system. This need is dictated by a problem arising where an implementation of a user interface relies on a set of closely coupled functions between the commands defined and the application functions or services of the embedded system. (Specification, p. 2, ¶¶ 6-7.)

In order to decouple the CLI from the services and features delivered and implemented in the embedded system, a CLI description language is implemented to define a set of command keywords and arguments forming CLI description files. (Abstract.) The CLI description files are compiled to create run time modules. The runtime modules, in turn, are provided as an application from the embedded system to interface applications. (Specification, p. 8, ¶ 30.)

Claim 1 is presented below with elements read on the figures of the drawings as suggested in MPEP 1206.

1. A method, including defining a set of commands to be used with a command line interface (CLI) (Fig. 12, block 1205; Specification, p. 26, ¶ 68), each command in the set of commands specifying an action to be performed in a system; defining a set of system interfaces (Fig. 12, block 1210; Specification, p. 26, ¶ 69) including objects and methods, wherein each action specified in the command is associated with an object and a method, the object and the method performing semantics required by the command; and creating a CLI description file (Fig. 12, block 1215; Specification, p. 26, ¶ 70) for each command in the set of commands to abstract the CLI from the system, the CLI description file mapping the command to be used with the CLI with the action and the associated objects and the associated methods (Specification, p. 14, ¶ 45), wherein the command is to be entered into the system in text mode.

As stated in MPEP 1206, the claims are not to be limited to this embodiment by such reading.

B. Summary of Rejections

Claims 1-29 stand rejected under 35 U.S.C. §102(b) as allegedly being anticipated by Bach et al.

(US patent no. 6,141,660).

C. Summary of the Reference

Bach is directed at a method, apparatus, and article of manufacture for generating class specifications for an object-oriented application that accesses a hierarchical database is disclosed. The class specifications are generated using a command line interface of a class definition tool. A database description and a record layout associated with the hierarchical database are captured and associated to define a specification for the database. Class definitions are then generated from the database specification, wherein the class definitions are instantiated as objects in the objects framework that encapsulate data retrieved from the database. (Bach, Abstract.)

VI. ISSUES

- A. Whether the 35 U.S.C. 102(b) rejection of claims 1-29 was erroneous, because Bach fails to teach each and every limitation of independent claims 1, 12, and 23?
 1. Whether Bach fails to teach the feature of “creating a CLI description file for each command in the set of commands to abstract the CLI from the system” as required by claim 1, to substantiate the 35 U.S.C. 102(b) rejection of claims 1-29?
 2. Whether the language in Bach relied upon by the office action, relating to objects framework and to graphical user interface (GUI), is unrelated to the feature of “creating a CLI description file for each command in the set of commands to abstract the CLI from the system” of claim 1, 12, and 23?

VII. GROUPING OF CLAIMS

For the purposes of this appeal, claims 1-29 stand or fall together.

VIII. ARGUMENT

A. *The 35 U.S.C. 102(b) rejection of claims 1-29 was erroneous, because Bach fails to teach each and every limitation of independent claims 1, 12, and 23*

To anticipate a claim, the reference must teach every element of the claim. “A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

1. *Bach is silent regarding a CLI description file for each command in the set of commands, and therefore fails to teach the feature of “creating a CLI description file for each command in the set of commands to abstract the CLI from the system” as required by independent claims 1, 12, and 23*

Claims 1, 12, and 23, include the following feature:

creating a CLI description file for each command in the set of commands to abstract the CLI from the system, the CLI description file mapping the command to be used with the CLI with the action and the associated objects and the associated methods, wherein the command is to be entered into the system in text mode.

Bach discloses the following:

The class specifications are generated using a command line interface of a class definition tool. The class definition tool parses database files and generates class definitions for objects that encapsulate or wrapper data retrieved from the database. The class definition tool also automatically generates input forms and output pages (for example, HTML or XML forms and pages) that are displayed on web browsers that interact with the application program and objects framework.

Bach, col. 5, ll. 37-45.

Bach fails to disclose or even suggest a feature of “**creating a CLI description file for each command in the set of commands to abstract the CLI from the system**”, the CLI description file mapping the command to be used with the CLI with the action and the associated objects and the associated methods, wherein the command is to be entered into the system in text mode”, as required by claims 1, 12, and 23.

Although Bach discloses operation and high level logic of a command line interface (CLI) (Bach, col. 16, l. 62 – col. 20, l. 51), Bach fails to disclose or even suggest a CLI description file mapping the command to be used with the CLI with the action and the associated objects and the associated methods. In Bach, there is no mention of a CLI description file, and no indication of a mechanism to abstract the CLI from the system. In contrast, the CLI in Bach may be relying on a set of closely coupled functions between the commands defined and the class definition tool, which is distinct and from and in stark contrast with “**creating a CLI description file for each command in the set of commands to abstract the CLI from the system**”, as required by claims 1, 12, and 23.

Because Bach does not disclose each and every element of claims 1, 12, and 23, claims 1, 12, and 23, as well as their respective dependent claims, are patentable and should be allowed.

2. **The language in Bach relied upon by the office action relates to objects framework and to graphical user interface (GUI), and thus is unrelated to the feature of “creating a CLI description file for each command in the set of commands to abstract the CLI from the system” of claims 1, 12, and 23**

The Detailed Action relies on “Bach, col. 8, ll. 1-23” and “Bach, col. 12, ll. 1-19” to show the abovementioned feature of “creating a CLI description file for each command in the set of commands to abstract the CLI from the system” of claim 1 (Official Paper 9, p. 2). However, the passage in “Bach, 8: 1-23” relates to objects framework, which resides on the server 102 (Bach, Fig. 1) and is *unrelated to the class definition tool (CDT) or its CLI, both of which reside on the client 100* (Bach, Fig. 1, Fig. 4). Furthermore, the passage in “Bach, col. 12, ll. 1-19” relates to a Graphical User Interface (GUI), as opposed to Command Line Interface (CLI), and therefore is not relevant to the feature of “**creating a CLI description file for each command in the set of commands to abstract the CLI from the system**”, the CLI description file mapping the command to be used with the CLI with the action and the associated

objects and the associated methods, wherein the command is to be entered into the system in text mode" of claims 1, 12, and 23. Thus, Bach does not disclose each and every element of claims 1, 12, and 23, and therefore claims 1, 12, and 23, as well as their respective dependent claims, are patentable and should be allowed.

IX. CONCLUSION

For the foregoing reasons, Appellants respectfully request reversal of the Examiner's rejections as set forth in the Final Office Action and request that the Board direct allowance of claims 1-29. If there are any additional charges, please charge Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 7-21-04, 2004

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025
(408) 720-8598



Elena B. Dreszer
Reg. No. 55,128



APPENDIX A
(37 C.F.R. § 1.192 (c)(9))

The claims on appeal read as follows:

1. (Previously Presented) A method, including:

defining a set of commands to be used with a command line interface (CLI), each command in the set of commands specifying an action to be performed in a system; defining a set of system interfaces including objects and methods, wherein each action specified in the command is associated with an object and a method, the object and the method performing semantics required by the command; and creating a CLI description file for each command in the set of commands to abstract the CLI from the system, the CLI description file mapping the command to be used with the CLI with the action and the associated objects and the associated methods, wherein the command is to be entered into the system in text mode.
2. (Previously Presented) The method of claim 1, wherein defining the set of commands includes defining keywords, arguments, input and output requirements for each command.
3. (Previously Presented) The method of claim 2, wherein defining the keywords and the arguments for each command includes defining help texts for the keywords and for the arguments.
4. (Previously Presented) The method of claim 2, wherein defining the input requirements includes defining an argument set for the command, and wherein defining the output requirements includes defining a display format to display a result responsive to executing the command.

5. (Previously Presented) The method of claim 4, wherein the argument set is associated with the action of the command, and wherein the argument set is included in an input string specified with the command, the argument set including zero or more arguments.
6. (Original) The method of claim 5, wherein an argument type is selected for each argument, and wherein each argument type is implemented as a Java class.
7. (Previously Presented) The method of claim 1, wherein defining a set of interfaces including objects and methods includes defining a Java class for the action, the Java class for the action naming the object associated with the action and having methods that can be invoked when performing the action.
8. (Previously Presented) The method of claim 7, further including registering the object with the system such that when the action is performed, the methods are invoked.
9. (Original) The method of claim 1, wherein the CLI description file is created using a generalized markup language.
10. (Original) The method of claim 9, wherein the generalized markup language is Extensible Markup Language (XML).
11. (Previously Presented) The method of claim 10, further including: compiling the CLI description file to generate a run time module.
12. (Previously Presented) A computer readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method including:

defining a set of commands to be used with a command line interface (CLI), each command in the set of commands specifying an action to be performed in a system; defining a set of system interfaces including objects and methods, wherein each action specified in the command is associated with an object and a method, the object and the method performing semantics required by the command; and creating a CLI description file for each command in the set of commands to abstract the CLI from the system, the CLI description file mapping the command to be used with the CLI with the action and the associated objects and the associated methods, wherein the command is to be entered into the system in text mode.

13. (Previously Presented) The computer readable medium of claim 12, wherein defining the set of commands includes defining keywords, arguments, and input and output requirements for each command.
14. (Previously Presented) The computer readable medium of claim 13, wherein defining the keywords and the arguments for each command includes defining help texts for the keywords and for the arguments.
15. (Previously Presented) The computer readable medium of claim 13, wherein defining the input requirements includes defining an argument set for the command, and wherein defining the output requirements includes defining a display format to display a result responsive to executing the command.
16. (Previously Presented) The computer readable medium of claim 15, wherein the argument set is associated with the action of the command, and wherein the argument set is included in an input string specified with the command, the argument set including zero or more arguments.

17. (Original) The computer readable medium of claim 16, wherein an argument type is selected for each argument, and wherein each argument type is implemented as a Java class.
18. (Previously Presented) The computer readable medium of claim 12, wherein defining a set of interfaces including objects and methods includes defining a Java class for the action, the Java class for the action naming the object associated with the action and having methods that can be invoked when performing the action.
19. (Previously Presented) The computer readable medium of claim 18, further including registering the object with the system such that when the action is performed, the methods are invoked.
20. (Original) The computer readable medium of claim 12, wherein the CLI description file is created using a generalized markup language.
21. (Original) The computer readable medium of claim 20, wherein the generalized markup language is Extensible Markup Language (XML).
22. (Previously Presented) The computer readable medium of claim 12, further including: compiling the CLI description file to generate a run time module.
23. (Previously Presented) A method, including:
defining a set of commands to be used with a command line interface (CLI);
defining keywords, arguments, input and output requirements for each command, each command in the set of commands specifying an action to be performed in a system;

defining a set of interfaces including objects and methods, wherein each action specified by the command is associated with an object and a method, the object and the method performing semantics required by the command;

creating a CLI description file for each command in the set of commands to abstract the CLI from the system, the CLI description file mapping the command to be used with the CLI with the action and the associated objects and the methods, wherein the CLI description file is created using Extensible Markup Language (XML); and

compiling the CLI description file to create a run time module,

wherein the command is to be entered into the system in text mode.

24. (Previously Presented) The method of claim 23, wherein defining the keywords and the arguments for each command includes defining help texts for the keywords and for the arguments.
25. (Previously Presented) The method of claim 23, wherein defining the input requirements includes defining an argument set for the command, and wherein defining the output requirements includes defining a display format to display a result responsive to executing the command.
26. (Previously Presented) The method of claim 25, wherein the argument set is associated with the action of the command, and wherein the argument set is included in an input string specified with the command, the argument set including zero or more arguments.
27. (Original) The method of claim 26, wherein an argument type is selected for each argument, and wherein each argument type is implemented as a Java class.
28. (Previously Presented) The method of claim 23, wherein defining a set of interfaces

including objects and methods includes defining a Java class for the action, the Java class for the action naming the object associated with the action and having methods that can be invoked when performing the action.

29. (Previously Presented) The method of claim 28, further including registering the object with the system such that when the action is performed, the corresponding methods are invoked.